# バイオインフォマティクスにおける基本アルゴリズム(数学者のための分子生物学入門,研究会報告)

AUTHOR(S):

阿久津, 達也; 山西, 芳裕

CITATION:

阿久津, 達也 ...[et al]. バイオインフォマティクスにおける基本アルゴリズム(数学者のための分子生物学入門,研究会報告). 物性研究 2003, 81(1): 120-129

ISSUE DATE:

2003-10-20

URL:

http://hdl.handle.net/2433/97610

# バイオインフォマティクスにおける基本アルゴリズム

阿久津　達也
京都大学化学研究所
ノート作成：山西芳裕（京都大学化学研究所）

## 1. はじめに

理論的計算科学や数学などの分野においては未解決な問題（Open Problem）が研究の進展のための大きな原動力となってきた。ゲノム情報科学におけるアルゴリズム的側面を扱った教科書によると、Pevzner と Waterman は 1995 年にゲノム情報科学における 57 個の未解決問題を提示したが、その約 1/4 が 2000 年までに解決されたと書いてある。また、この本には 117 個の問題が列挙されている。そのうち、いつくが解決済みかは明記されていないが、半分程度は未解決なのではないかと思われる。本セミナーでは、ゲノム情報科学における様々なアルゴリズムを紹介するとともに、課題や問題点を指摘する。とくに、ホモロジー検索、進化系統樹推定、RNA 二次構造予測、タンパク質立体構造予測についての説明を行う。

## 2. ホモロジー検索

### 2.1. ペアワイズアライメント

配列アライメントはバイオインフォマティクスの最重要技術の一つであり、2 個もしくは 3 個以上の配列の類似性の判定に利用される。文字間の最適な対応関係を求める最適化問題に帰着し、配列長を同じにするようにギャップ記号（挿入、欠失に対応）を挿入していく。ところで、ゲノム情報科学において、最適アライメント問題は一言で言うと、二個（以上の）文字列が与えられたときに、文字間の対応関係を求める問題、つまり、文字列パターンマッチング問題の一種である。2 個の DNA 配列もしくは、アミノ酸配列の類似性を判定するには、このアライメントを計算することが必須であり、理論的観点からも実用的観点からもさまざまな研究が行われてきた。

DNA 配列を例にとり、2 個の文字列のアライメント問題を定義する。ここでは DNA 配列を考えているので、各文字列は、A,C,G,T の 4 種類の文字列から構成される。この 4 種類の文字の各ペアの間にはスコア（類似度）が定義されるが、ここでは簡単のために同じ文字（X) 間のスコアは 1 とし（s{X,X}=1)、異なる文字（X,Y) 間のスコアは－1 としよう。ここで、2 個の文字列が与えられた時に、各文字列にギャップ記号（－）を挿入して、2 個の文字列の長さが同じになるようにする。たとえば、AGCT と TCGCT の場合には、

$$A-GCT$$

$$TCGCT$$

というようにギャップ記号を入れれば、長さが同じになる。このようにギャップを入れて長さを同じにしたものは「アライメント」と呼ばれ、同じ列の文字列どうしが対応しているものとみなされる。そして、列ごとのスコアを加算していくことにより、アライメント全体のスコアの計算することができる。上の例だと、s{A,T} + s{-,C} + s{G,G} + s{C,C} + s{T,T} がスコアであり、その和は 1 となる。ただし、ここでは簡単のため、

ギャップ文字と他の文字の間のスコアは常に-1であるものとした（s{X,-}=s{-,X}=-1）。なお、アライメントにおいてはギャップ記号どうしが同じ列に並んではいけないものとする。ギャップペナルティの定義の仕方も様々であり、線形コスト $-gd$ やアフィンギャップコスト $-d-e(g-1)$ がある。ここで、$g$ はギャップ長、$d$ はギャップ（開始）ペナルティ、$e$ はギャップ伸張ペナルティである。

アミノ酸配列のアライメントにおいては、残基間（アミノ酸文字間の）が完全に一致するかどうかだけ考えるのではなく、残基間の類似性を考慮に入れる。類似性を表す行列として、PAM250、BLOSUM45などのスコア行列（置換行列）を利用する。つまり、アミノ酸は20種類あるので、20x20の類似行列に対応する。スコア行列は、どのように導出されるかというと、基本的には頻度の比の対数をスコアとする。たとえば、BLOSUM行列の作り方を説明する。1）既存のスコア行列を用いて多くの配列のアライメントを求めギャップなしの領域（ブロック）を集める。2）残基がL%以上一致しているものを同一クラスターに集める。3）同じクラスタ内で、残基aが残基bにアラインされる頻度 $A_{ab}$ を計算する。4）$q_a = \sum_b A_{ab} / \sum_{cd} A_{cd}, p_{ab} = A_{ab} / \sum_{cd} A_{cd}$ を求め、$s(a,b) = \log(p_{ab}/q_a q_b)$ としたのち、スケーリングし近傍の整数値に丸める。

最適アライメント問題を解くためのアルゴリズムとして、すべてのアライメントを作り、それぞれのスコアを計算し、最大スコアのアライメントを求めるというアルゴリズムが考えられる。残念ながら、このアルゴリズムは非常に効率の悪い（指数オーダーの）アルゴリズムとなってしまう。というのも、配列が2個の場合のペアワイズアライメントの場合でも、可能なアライメントの個数は指数オーダーであることが知られているからである。そこで、効率のよいアルゴリズムが必要となるが、20年以上昔より動的計画法という一般的な手法に基づくアルゴリズムが知られている。

ここでは、そのアルゴリズムを簡単に説明する。このアルゴリズムでは、アライメント問題を図2のような2次元のメッシュ上のダイアグラム（有向グラフ）における最長経路問題に変換する。この図では、横方向が1個目の文字列に対応している。また、右下に向かう各矢印には、対応する文字間のスコアが対応付けられている。また、横方向の矢印および縦方向の矢印にはギャップ文字1文字分のスコアが対応づけられている。すると、アライメントと、左上の点から右下の点までのパス（経路）には、1対1対応が存在し、また、パス上のスコアの和は対応するアライメントのスコアに等しくなる。よって、長さ（スコアの和）が最長となる経路を求めればよい。一般に最長経路問題は難しい問題であるが、図2のグラフは規則的で単純な形をしているため、動的計画法という手法により比較的効率よく計算することができる。アミノ酸配列の場合も同様であり、残期間のスコアの計算が対応するスコア行列の要素を用いて行われる。つまり重み付のパス計算でアライメントスコアを計算することになる。

実際、配列の一部のみ共通部分があることが多く、共通部分のみのアライメントが求められることが多い。$x_1, x_2, \cdots, x_m, y_1, y_2, \cdots, y_n$ を入力とするとき、スコアが最大となる部分列ペア $x_i, x_{i+1}, \cdots, x_k, y_j, y_{j+1}, \cdots, y_h$ を計算するのが目的である。たとえば、HEAWGEHとGAWEDの場合、

$$AWGE$$

$$AW-E$$

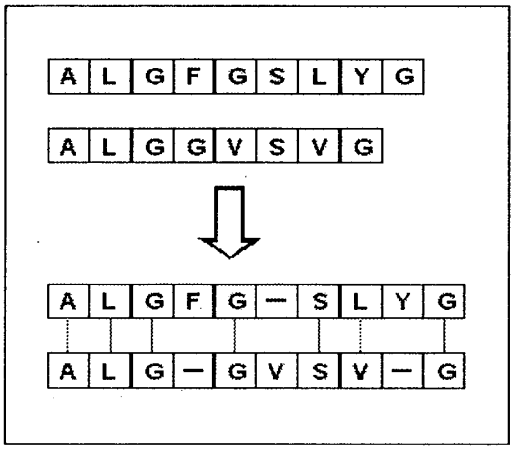


図 1. アミノ酸配列のペアワイズアライメントの例

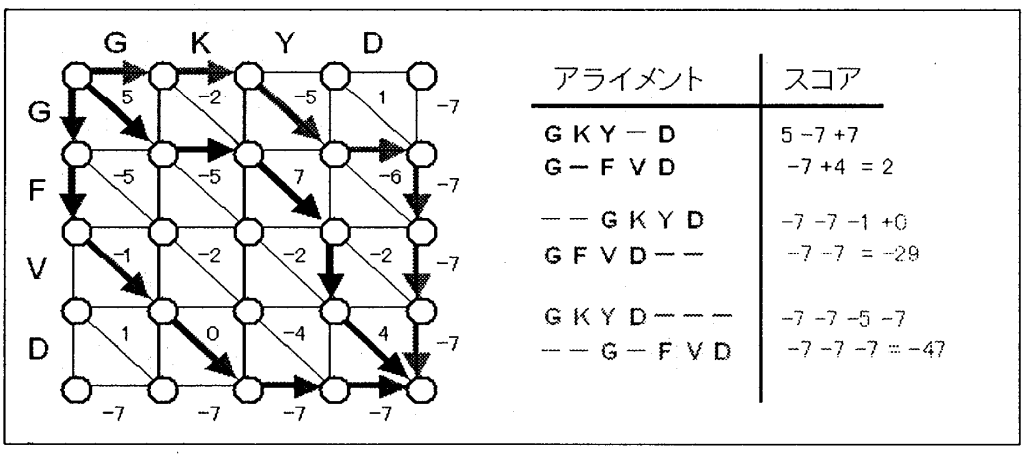


図 2. 最長経路問題

というアライメントを計算する。大域アライメントを繰り返すと、$O(m^3n^3)$ 時間であるが、Smith-Waterman アルゴリズムなら $O(mn)$ 時間で解くことができる。

この計算時間や必要メモリについて簡単な計算をしてみる。2 個の入力文字列の長さをそれぞれ、n,m とする。すると、図 1 のように作られるグラフには、$(n+1)\times(m+1)$ 個の点が存在することになるため、計算時間やメモリも $(n+1)\times(m+1)$ 程度となる（$O(nm)$ 時間、$O(mn)$ スペース）。n と m が同じくらいとすると、これは $O(n^2)$ 時間、$O(n^2)$ スペースのアルゴリズムとなる。実はこれまでの議論はそれほど難しいことではない。ところが、もっと効率のよいアルゴリズムはあるか？ちうことになると、とたんに難しい未解決問題となってしまう。$O(n^2)$ 時間より本質的に効率のよいアルゴリズムは知られておらず、また、$O(n^2)$ 時間が最適である、という結果も知られていない。ただし、メモリ効率に関しては、$O(n)$ のアルゴリズムがかなり昔より知られている。

$$F(0,j) = -jd, \quad F(i,0) = -id$$

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases}$$

図 3. グローバルアライメントの動的計画法

$$F(i,j) = \max \begin{cases} 0 \\ F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases}$$

図 4. ローカルアライメントの動的計画法

### 2.2. FASTA、BLAST

これまでは、2個の文字列の比較を考えてきたが、実際には、一個の配列を入力し、データベース中の数百万の配列とのアライメントを計算し、最適なものを一個（もしくは数十個程度）もとめるというように利用される場合が多い。この場合、各配列の長さをn程度、データベース中の配列の個数をNとし、上記の動的計画法に基づくアルゴリズムを繰り返し適用すると $O(n^2N)$ 時間かかることになる。この問題に関しては、FASTA や BLAST、SSEARCH、PSI-BLAST などのハッシュ法に基づく実用的なアルゴリズムが開発され、生物学者らによって実用的に使用されるにいたっている。FASTA は、短い配列（アミノ酸の場合、1-2 文字、DNA の場合、4-6 文字）の完全一致をもとに対角線を検索し、さらにそれを両側に伸張し、最後に DP を利用する。BLAST は、固定長（アミノ酸の場合では 3、DNA の場合では 11）の全ての類似語単語のリストを生成し、ある閾値以上の単語ペアを探し、それをもとに両側に伸張する。ギャップは入らない。伸張の際に極値分布に基く統計的な値 E-value を利用する。SSEARCH では、局所アライメント（Smith-Waterman アルゴリズム）をそのまま実行する。PSI-BLAST では、ギャップを扱えるように拡張した BLAST を繰り返し実行する。「BLAST で見つかった配列からプロファイルを作り、それをもとに検索」という作業を繰り返す。しかしながら、本質的に $O(n^2N)$ より効率のよいアルゴリズムは知られておらず、この改善は未解決問題であると思われる。また、ギャッ

プを許したマッチのためのハッシュ関数で理論的保証があり、かつ実用的なものは知られていないと思われる。

### 2.3. マルチプルアライメント

2個の文字列に対するアライメントは$O(n^2)$時間でできたが、k個の文字列に対するアライメントはk次元のメッシュ上グラフ上の最長経路問題に変換することにより、$O(n^k)$時間で計算できることが知られている。このアルゴリズム自体は大学院入試レベルのアルゴリズムである。しかしながら、この計算量はkがちょっとでも増えると現実的でなくなるので、より効率のよいアルゴリズムを開発することが求められる。すると、それはとたんに難しい未解決問題となり、現在のところ本質的に$O(n^k)$時間より効率のよいアルゴリズムは知られていないということになる。また、kが固定されていない場合には、一般にNP困難という難しいクラスの問題になることも知られている。

厳密な最適解を求めるのはNP困難であるため、実用的なマルチプルアライメントを実行するためには、ヒューリスティックなアルゴリズムの開発が求められる。例えば、ひとつの方法として、プログレッシブアライメントがある。1）近隣結合法などを用いて、案内木を作る。2）類似度が高い節点から低い節点へという順番で、配列対配列、配列対プロファイル、プロファイル対プロファイルのアライメントを順次計算する。もう一つは、逐次改善法と呼ばれる方法で、「配列を一本取り除いてはアライメントしなおす」を繰り返すというものである。

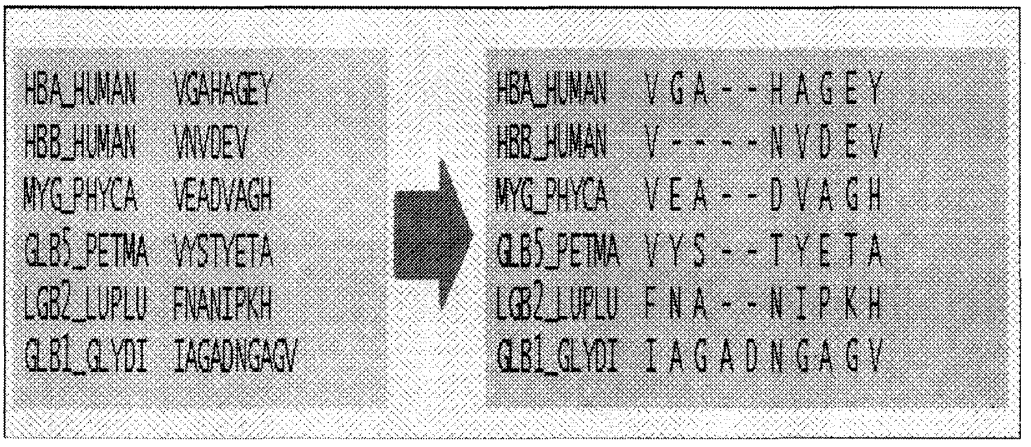


図 5. アミノ酸配列のマルチプルアライメントの例

確率論的なアプローチとして、隠れマルコフモデル（HMM）がある。HMMとは、有限オートマンと確率を組み合わせたものである。HMMにおける基本アルゴリズムとして、出力記号から状態列を推定するViterbiアルゴリズムと、出力記号列からパラメータを推定するBaum-Welchアルゴリズム（EMアルゴリズム）がある。応用としては、配列をアライメントするのに使われるわけだが、状態列として、一致状態（M）、欠失状態（D）、挿入状態（I）を持つHMMは、プロファイルHMMと呼ばれている。

NP困難な最適化問題があった場合に精度保証のある近似アルゴリズムを研究開発するというのは情報科学における一般的な研究スタイルであるが、この問題については、ある

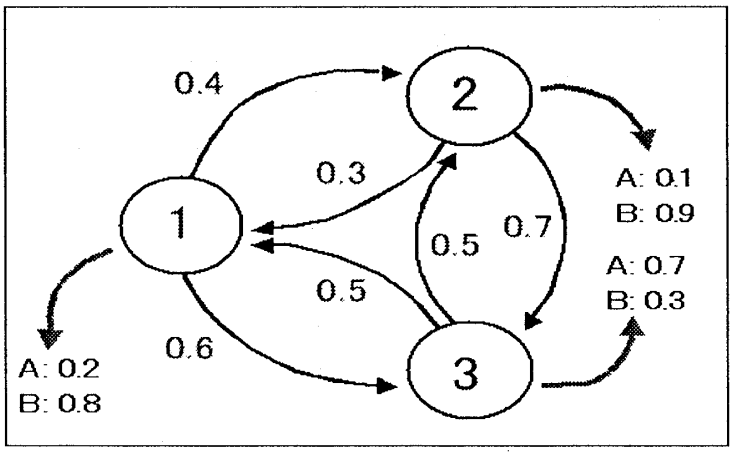


図 6. 隠れマルコフモデル

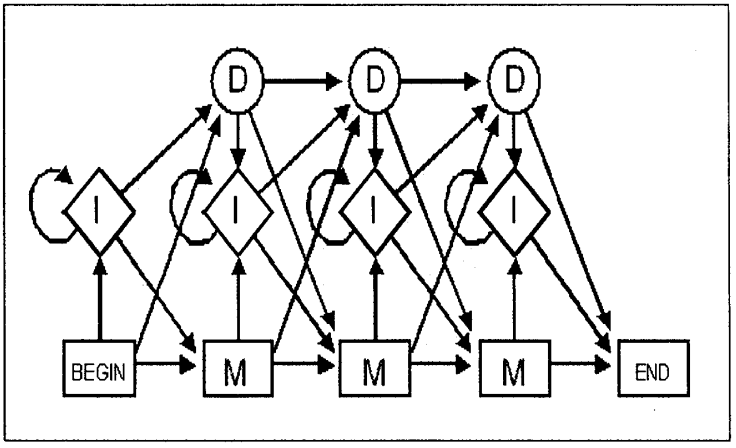


図 7. プロファイル HMM

定式化の下で近似率 2 というアルゴリズムが 1993 年に発表された。しかしながら、些細な改良はあるものの、これまで本質的に 2 という近似率は改良されていない。これも有名な未解決問題である。なお、2000 年に、特殊な場合に限っては近似率を 1（最適解）にいくらでも近づけることのできるアルゴリズム（PTAS）が、コロモゴルフ計算量などでも有名な Ming li らによって開発された。このように、これまで多くの研究が行われてきたアライメントに限っても、情報科学的観点から見れば様々な未解決問題があることが分かる。

### 3. 進化系統樹推定

生物種の進化における系統関係は、系統樹と呼ばれる樹状の図で表現される。これはクラスタリングなどにも関係する問題であり、複数の生物種の遺伝子配列が与えられた時に、その配列を比較し、進化の過程を表す木を再構成する。比較したい複数の生物種間で共通に保存されているオーソログ遺伝子を同定し、その配列情報からマルチプルアライメ

ントを作ることから始める。塩基配列やアミノ酸配列といった分子レベルの情報から分子系統樹を推定する方法として、距離行列法や近隣結合法がある。

距離行列法の一つであるUPGMA法では、まずn本の配列を独立のクラスターとみなし、隣接したクラスターを順次まとめていく。クラスター間の非類似度として、それぞれのクラスターに含まれる配列全ての組み合わせのの平均距離で定義する。つまり、多変量解析法のクラスター分析のひとつの方法である平均距離法のアルゴリズムを用いるわけである。最後に全体がひとつのクラスターとなったときのデンドログラムが系統樹となる。これに対して、配列間の距離がある制約を満たす場合には、近隣結合法というアルゴリズムにより木が正しく再構成されることが知られている。

近隣結合法では、最初にn個の配列を外部ノードとして、ひとつの内部ノードにつないだ星型のトポロジーから出発し近隣のクラスターを順次くくりだしていく。どのノード対をくくりだすかは、全ての組み合わせを考え、エッジの長さが最小になるものを選ぶ。くくりだされた部分は新たなクラスターとなり、距離行列のサイズがひとつずつ減っていくことになる。しかしながら、より一般の距離や、また、最節約法といった基準を用いた場合には、再構成が困難になることが知られている。

UPGMA法では進化速度が一定であるという条件でエッジの長さの総和を最小化しているのに対して隣接結合法では進化速度が一定でないことを考慮して各ステップで最小化の基準を用いている。最適解を計算する実用的なアルゴリズム、もしくは、精度保証のある近似アルゴリズムを開発することが望まれるが、決定版と呼べるようなアルゴリズムは知られていない。

## 4. RNA二次構造予測

1本鎖RNAの二次構造は、相補的な配列が局所的に二重らせん構造をとることにより形成される。二重らせん部分をステム、ステムをつなぐ一本鎖の部分をループと呼ぶ。RNA二次構造予測とは、どの塩基対が結合しているかを予測する問題である。方針として、ステム構造と各ループ構造の安定性の評価値を合計したスコア関数を最大化することによって行われる。ベースペアとして、B={{a,u},{g,c}}を考え、RNAの二次構造Mは、

$M = \{(i,j)|1 \leq i < j \leq n, \{a_i, a_j\} \in B\}$かつ$i \leq h \leq j \leq k$を満たす$\{a_i, a_j\}, \{a_h, a_k\} \in M$

は無いという条件を満たすとする。スコア関数を

$$\mu(a_i, a_j) = 1 \quad if \{a_i, a_j\} \in B$$

$$\mu(a_i, a_j) = 0 \quad otherwise$$

と定義したとき、

$$\sum_{(i,j) \in M} \mu(a_i, a_j)$$

が最大になるようなMを選択する。

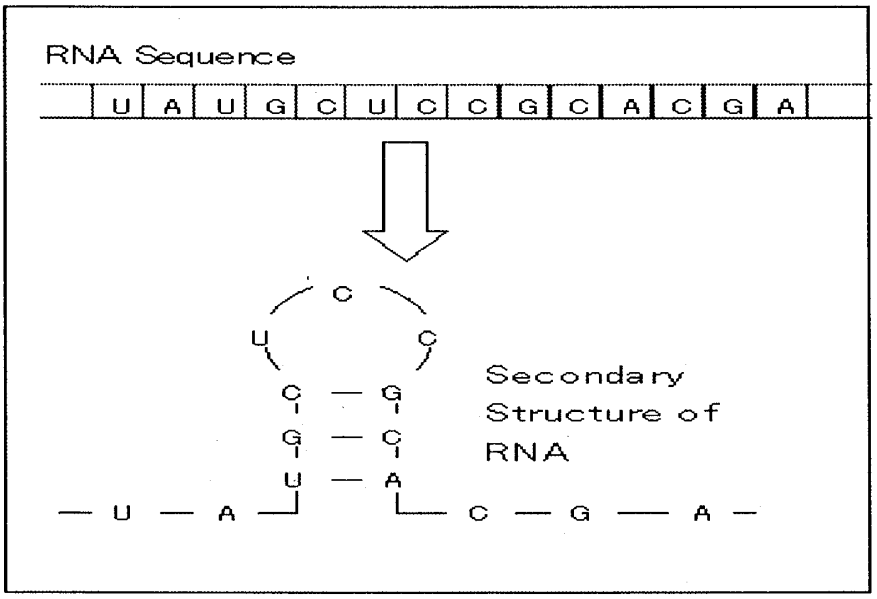


図 8. RNA の 2 次構造予測

## 5. タンパク質立体構造予測

タンパク質立体構造予測とは、アミノ酸配列からタンパク質の立体構造（3 次元構造）をコンピュータにより推定することである。実験よりもはるかに精度は悪いが、大体の形が分かればよいのであれば、5 割近くの精度で予測できる。タンパク質は主に、球状タンパク質、繊維状タンパク質、膜タンパク質の 3 つに分類される。構造の視点では、一次構造（アミノ酸配列）、二次構造（$\alpha$、$\beta$、それ以外）、三次元構造（三次元構造、立体構造）、四次構造（複数の鎖）に分類される。タンパク質立体構造の決定は、主に X 線結晶解析か NMR 解析による。一般に X 線解析のほうが精度が高いが、結晶中の構造しか分からない。アミノ酸配列決定よりも困難な作業で、半年から 1 年ぐらいかかることも珍しくなく、既知の立体構造が少ない理由となっている。タンパク質の立体構造は、らせん状の $\alpha$ ヘリックス、ひも状の部分が並んだ $\beta$ シートの二種類の特徴的な構造が頻繁に現れ、立体構造のコアを作る。これらの特徴を持つ構造をコンピュータを使って予測してやる方法は、物理学的原理に基づく方法、格子モデル、2 次構造予測、スレッディングの 4 つに分類できる。

物理的原理に基づく方法は、エネルギー最小化、もしくは微分方程式を数値的に解く、などの物理的原理に基づく。主として分子動力学法が使われ、数十残基程度であれば、実際のたんぱく質やペプチドと似た構造を推定可能なことがある。格子モデルに基づく方法では、各残基が格子点にあると仮定される。予測よりもフォールディングの定性的な理解のために利用される。ストリングフォールディング問題とよばれ、タンパク質立体構造予測問題を極端に単純化した問題であり、二進文字列（0 と 1 からなる文字列）をスコアが最小になるように平面格子（もしくは立方格子）上に埋め込むという問題であり、エネルギーの最小化に対応する。2 次元で 1/4 近似、3 次元で 3/8 近似できることが指摘されている。またこれまでの研究によって、3 次元で NP 困難、2 次元で NP 困難、2 次元で 1/3

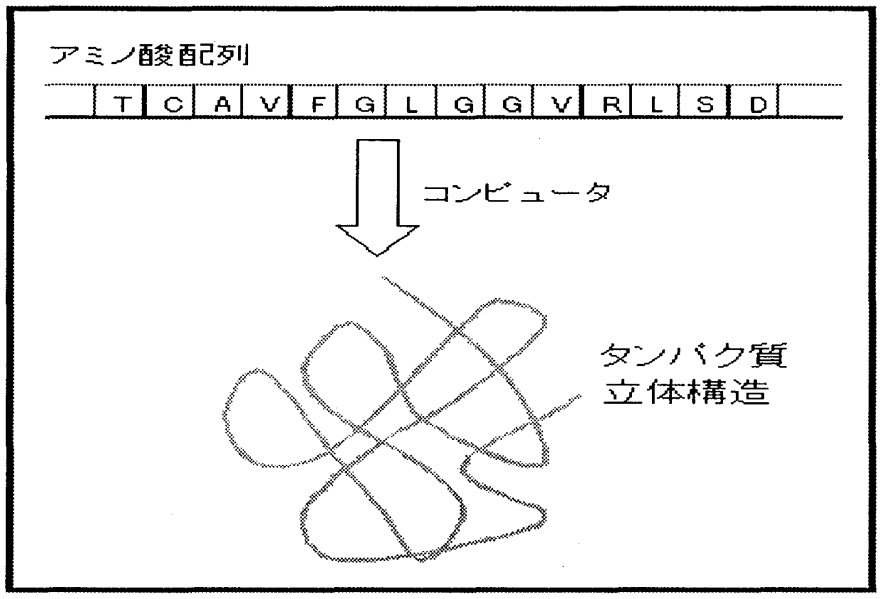


図 9. タンパク質の立体構造予測

近似できることが指摘されているが、さらなる近似は課題となっている。

二次構造予測では、アミノ酸配列中の各残基が、$\alpha$、$\beta$、それ以外のどれかに属するかを予測するという問題である。高精度なソフトだと 70 から 80 %の予測率であり、ニューラルネット、HMM、SVM などが利用されている。スレッディングによる構造予測は、以下の 3 つのステップから実行される。構造未知の配列と既知の立体構造（数百種類程度）の間のスレッディングを、それぞれ計算する。スレッディング結果のスコア（適合度）が最も高い構造を採用（スレッディングにより対応付けられた座標にアミノ酸を配置）する。必要があれば、分子動力学法などを用いて構造を最適化する。スレッディング法には大きく分けて、プロファイルによるスレッディング（PSI-BLAST、3D-1D 法、構造アライメント結果に基づくスレッディング）と、残基間ポテンシャルによるスレッディング（コンタクトポテンシャル、距離依存ポテンシャル、その他のポテンシャル）がある。プロファイル法では、動的計画法により最適解を計算する過程で、スコア行列のかわりにプロファイルを使う。よく使われるのは、3D-1D プロファイルであり（1991 年 Eisenberg によって提案）、構造中の残基を 18 種類の環境（3 種類の二次構造、6 種類の内在性や極性の組み合わせを考える）に分類し、それに属するスコアをベクトルにしたものとして定義される。

ポテンシャル型スコア関数を用いたスレッディングでは、全体のポテンシャルエネルギーを最小化する、すなわち $\sum f_d(X,Y)$ を最小となるようなスレッディングを計算する。ポテンシャル関数を用いた場合、厳密な最適解の計算は困難で NP 完全であることが知られている。アルゴリズムとして様々な方法が提案されている。分枝限定法は、コア領域内でのギャップは許さないが、多くの場合現実的な時間で最適解を計算可能である。Frozen 近似は、通常の動的計画法のアルゴリズムの応用版である。Double 動的計画法は、通常の動的計画法を二重に用いる方法で、立体構造アライメントなどにも応用可能である。

立体構造予測コンテスト（CASP）は、ブラインドテストにより予測方法を客観的に評

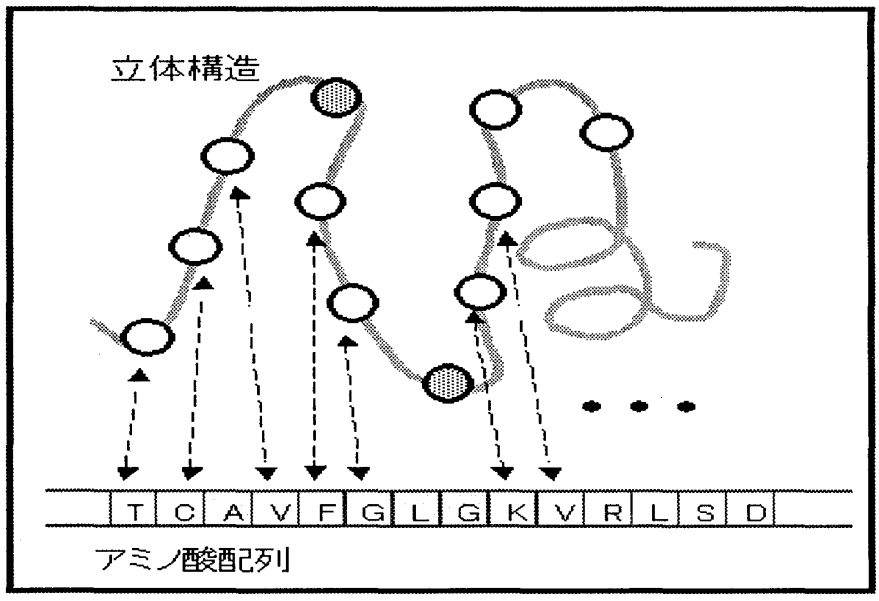


図 10. スレッディング

価しようとするための企画であり、誰でも参加可能である。ブラインドテストにより予測方法を評価する。データは、半年以内に立体構造が実験により決定する見込みの配列（数十種類）をインターネットで公開される。参加者は予測結果を送付する。構造決定後、正解とのずれなどを評価し、順位付けを行う。

立体構造予測の分野では、スレッディングの発明が大きなブレークスルーとなった。これによって構造既知の配列と類似性がない配列の構造を予測できるようになった。また、PSI-BLAST（ギャップを考慮したBLAST）の開発によって、プロファイルに基くマルチプルアライメントの繰り返し実行によるスレッディングが可能となった。現在のところ、Daivid Bakerによる統計情報とシミュレーションを組み合わせた、ab initio 予測法が、良い結果を出しており注目されている。

## 6. まとめ

バイオインフォマティクスにおいても（アルゴリズム論駅観点から）未解決問題は多いが、新しい数学につながるかどうかは疑問である。配列検索などはかなり実用的となっているが、構造予測はあまり実用的とはいえない。実用上のブレークスルーは問題を解くことよりも、新たな問題や数理モデルを作ることによりもたらされることが多い。細胞や生体のシミュレーションでは良い数理モデルを作ることが特に重要だと思われる。